REMARKS

Claims 1-39 are pending in this application. Applicants amend claims 1, 3, and 18, and cancel claims 2 and 19, as indicated in the above listing of the claims. Support for the amendments can be found in the original claims. No new matter is added. The application is believed to be in condition for allowance. Hence, reconsideration and allowance are respectfully requested.

Rejections Under 35 U.S.C. 103

The Office Action rejects claims 1-5, 9-14, 16-20, 26, 27, 30-34, 36-39 as being obvious over U.S. Patent No. 6,301,252 of Rangachar in view of U.S. Patent No. 6,625,590 of Chen.

Claim 1, as amended, recites a method of managing a telecommunications network device that includes the steps of registering at least one command executable by an application with one of a plurality of distributed command proxies associated with a command interface, which is local to the application. The command is then registered through this command proxy with a central command daemon that is associated with the command interface. This allows that command, which is received at the command interface from a user interface, to be forwarded to the application that executes the command. Support for amendments to claim 1 can be found in the original claim 2 and throughout the remainder of the specification. Thus, no new matter is added.

Rangachar describes a method for centralized management and remote control of a network of cell-based switches. A network manager receives commands directed to switches and processes the commands in order to generate generic commands. The manager sends the generic commands to one or more of the switches. Each switch includes a vendor-specific command interface that maps the generic commands issued by the manager into instructions for controlling the hardware of a control unit of the switch.

Chen is directed to a command line interface for a network management platform. The command inputs received by the interface are processed by a parser that validates their syntax.

The interface also includes a command processor that executes the commands validated by the parser.

Neither Rangachar nor Chen teaches distributed command proxies. Nor do they teach registering a command executable by an application with a command proxy local to that application. The passage of Rangachar at col. 5, lines 23-28, to which the Examiner refers, does not teach command proxies but rather states that the network manager can carry out a network operation by creating a plurality of processes, one process for each of the switches, and exchanging information among the switches. There is, however, no indication that any of these processes would function as a *command proxy* through which a command executable by an application could be registered with a *central command daemon.*

Further, Chen does not cure the shortcomings of Rangachar in that it fails to teach command proxies with which commands executable by applications can be registered and through which these commands can then be registered with a central command daemon. The Examiner refers to a passage of Chen at col. 7, lines 55-59 to assert that Chen teaches registering at least one command executable by an application with a command interface. Applicants respectfully disagree for the following reasons. This passage of Chen simply states that a network environment in which Chen's command line interface can be utilized can store interface command files, which may be remotely invoked to perform management operation. This passage, however, does not teach registering applications that are capable of executing the commands with the command interface, much less performing such registration by utilizing a plurality of distributed command proxies in communication with a central command daemon.

Thus, the combined teachings of Rangachar and Chen fail to teach salient features of amended claim 1, and their concomitant advantages, such as the ability to maintain one set of code for each command regardless of which command interface (e.g., web, CLI, NMS) initiates the command. *See* specification, page 306.

Hence, claim 1 is patentable over the cited art. Claim 2 is canceled as its features are now incorporated in amended claim 1. Further, claims 3-10 depend either directly or indirectly on claim 1, and hence are also patentable.

Independent claim 11 recites a method of managing a telecommunications network device that includes the steps of registering at least one command executable by an application with a first command proxy, which is local to the application, and registering the command through the first command proxy with a central command daemon. The method further calls for receiving the command at a user interface and forwarding it to a second command proxy, which is local to the user interface. The command is then forwarded through the second command proxy to the central command daemon, and through the central command daemon to the first command proxy. This is followed by forwarding the command through the first command proxy to the application and completing execution of the command.

As discussed above, neither Rangachar nor Chen teaches utilizing command proxies in a network device, much less registering a command executable by an application with a central command daemon, via a first command proxy, and transmitting a command received by a user interface to the central command daemon, via a second command proxy.

Thus, claim 11 is patentable over the combined teachings of the references.

Independent claim 12 recites a method of managing a telecommunications network including a first network device and a second network device. The method calls for executing a community command daemon on one of the first or second network devices, executing a first application on the first network device and executing a second application on the second network device. A first command executable by the first application is registered with a first command interface on the first network device, and a second command executable by the second application is registered with a second command interface on the second network device. And the first and second commands are registered with the community command daemon.

Rangachar does not indicate that any applications running on the switches would register commands they are capable of executing with their respective command interfaces. Further, there is no indication in Rangachar that such commands would be registered with the network manager. Rather, the network manager generates generic commands that are parsed by the vendor-specific command interfaces of the switches. Chen does not bridge the gap in Rangachar's teachings in this regard as it fails to teach the command registrations recited in claim 12.

Hence, claim 12 and claims 13-17, which depend either directly or indirectly on claim 12, distinguish patentably over the teachings of Rangachar and Chen.

Independent claim 18 recites a telecommunication network device that includes an application capable of executing a command and a common command interface. The common command interface comprises a distributed system having a central command daemon and a plurality of distributed command proxies. The application is capable of registering the command with the common command interface and the common command interface is capable of receiving the command from a user interface and forwarding the received command to the application.

The arguments presented above apply with equal force to establish that claim 18 is also patentable. In particular, neither Rangachar nor Chen teaches a command interface having a central command daemon and a plurality of distributed command proxies. Hence, claim 18, and claims 19-27, which depend either directly or indirectly on claim 18, are patentable over the cited art.

Independent claim 28 recites a telecommunications network device that includes a common command interface, and an application that is capable of executing a command. The application includes a command application programming interface (API) for registering the command with the common command interface.

Neither Rangachar nor Chen teaches an application, running on a network device, that is capable of executing a command, and further includes a command application programming interface (API) for *registering the command with a common command interface*. For example, there is no indication in Rangachar that any of the applications executing on the switches are capable of registering commands that they are capable of executing with the central manager.

Hence, claim 28 and claim 29, which depends on claim 28, are patentable over the cited art.

Independent claim 30 recites a telecommunications network that includes a first network device and a second network device connected to the first network device. The network further includes a community command daemon executing on the first or the second network device. The network further includes a first common command interface that executes on the first network device and that is capable of registering a first command with the community command daemon. A second common command interface executes on the second network device and is capable of registering a second command with the community command daemon.

The arguments presented above apply with equal force to establish that claim 30, and claims 31-39 that depend directly or indirectly on claim 30, also distinguish patentably over the combined teachings of Rangachar and Chen.

In Paragraph 5, the Office Action rejects claims 6, 7, 8, 15, 21-25, 28, 29, and 35 as being unpatentable over Rangachar and Chen, in further view of U.S. Patent No. 6,664,978 of Kekic.

Claims 6, 7, and 8 depend on independent claim 1, claim 15 depends on independent claim 12, claims 21-25 depends on independent claim 18, claim 29 depends on independent claim 28, and claim 35 depends on independent claim 30. As discussed in detail above, Rangachar and Chen fail to teach salient features of these independent claims. Specifically, they do not teach the distributed command proxies of the claimed invention. In addition, Kekic does not remedy the deficiencies of Rangachar-Chen. In particular, Kekic does not teach distributed

command proxies. Rather, it is directed to a network management system having a managed element server for managing and monitoring a computer network.

As noted above, independent claim 28 is directed to a telecommunications network device that includes a common command interface, and an application that is capable of executing a command. The application includes a command application programming interface (API) for registering the command with the common command interface. As discussed above, neither Rangachar nor Chen teaches an API for registering commands with a common command interface. And this deficiency is not remedied by Kekic. The passage referred to by the Examiner in Kekic (col. 72, lines 34-50) suggests that the managed element server can be written in the JAVA programming language, and that a server API exists. There is, however, no mention in this passage of using an API to register commands.

Thus, the combined teaching of Rangachar-Chen and Kekic fail to teach salient features of claim 28. Hence, claim 28 is patentable over the cited art.

CONCLUSION

In view of the above amendments and remarks, Applicants respectfully request reconsideration and allowance of the application. Applicants invite the Examiner to call the undersigned at (617) 439-2514 if there are any questions.

Dated: March 18, 2005

Respectfully submitted,

By_____

Reza Mollaaghababa
Registration No.: 43,810
NUTTER MCCLENNEN & FISH LLP
World Trade Center West
155 Seaport Boulevard
Boston, Massachusetts 02210-2604
(617) 439-2000
(617) 439-9000 (Fax)
Attorney for Applicant